Optimisation bayésienne par méthodes SMC

Romain Benassi, Julien Bect, Emmanuel Vazquez

Optimisation bayésienne par méthodes SMC

Romain Benassi, Julien Bect & Emmanuel Vazquez

SUPELEC, Plateau de Moulon
3, rue Joliot-Curie,
91192 Gif-sur-Yvette cedex

prenom.nom@supelec.fr

Résumé. Le problème considéré est l'optimisation d'une fonction réelle f à l'aide d'une approche bayésienne. Les évaluations de f sont choisies séquentiellement à partir d'informations *a priori* sur la fonction f, modélisée par un processus aléatoire, et des évaluations précédentes. Cette approche présente deux problèmes, à savoir l'estimation des lois *a posteriori* de paramètres intervenant dans le choix des points d'évaluations, et la maximisation du critère utilisé pour déterminer ce choix. Dans cet article, nous proposons une approche SMC (*Sequential Monte Carlo*) pour résoudre ces deux problèmes de façon simultanée.

Mots-clés. optimisation globale, approche bayésienne, approche SMC

Abstract. We consider the problem of optimizing a real-valued continuous function f using a Bayesian approach, where the evaluations of f are chosen sequentially by combining prior information about f, which is described by a random process model, and past evaluation results. The main difficulties with this approach is to be able to compute the posterior distributions of quantities of interest which are used to choose evaluation points, and to maximize in an efficient way the criterion used to determine this choice. In this article, we decide to use a Sequential Monte Carlo (SMC) approach.

Keywords. Bayesian Global optimization, Computer experiments, Sequential design of experiments, Sequential Monte Carlo

1 Introduction

Le problème considéré est la recherche du maximum global d'une fonction $f : \mathbb{X} \to \mathbb{R}$, avec $\mathbb{X} \subset \mathbb{R}^d$, en utilisant le critère *Expected Improvement* ou EI [1, 3]. De nombreux exemples de la littérature mettent en évidence que les algorithmes fondés sur le critère EI sont particulièrement intéressants afin d'optimiser une fonction coûteuse à évaluer, ce qui est fréquemment le cas dans le domaine de l'analyse et de la conception de systèmes [2]. Néanmoins, passer du cadre théorique général tel que décrit dans [1] à une mise en œuvre informatique efficace est un problème difficile.

L'idée principale de l'optimisation à l'aide du critère EI est d'utiliser une approche bayésienne : f est vue comme une trajectoire d'un processus aléatoire ξ défini sur $\mathbb{R}^d$. Le processus ξ est généralement considéré gaussien conditionnellement à un paramètre $\theta \in \Theta \subseteq \mathbb{R}^s$, caractérisant ses fonctions moyenne et de covariance. Pour n résultats d'évaluation $\xi(X_1), \ldots, \xi(X_n)$ aux points $X_1, \ldots, X_n$, on note $\mathcal{F}_n$ la σ-algèbre générée par $X_1, \xi(X_1), \ldots, X_n, \xi(X_n)$, et $\mathbf{E}_{n,\theta}$ l'espérance conditionnelle à $\mathcal{F}_n$ et θ. On choisit un *a priori* π_0 sur θ, et on note π_n la distribution *a posteriori* conditionnellement à $\mathcal{F}_n$. Par la suite, après s'être donné initialement n_0 résultats d'évaluations, l'algorithme construit une suite de points d'évaluation $X_{n_0+1}, X_{n_0+2}, \ldots$ tel que, pour tous $n \geq n_0$,

$$X_{n+1} = \operatorname*{argmax}_{x \in \mathbb{X}} \bar{\rho}_n := \int_{\theta \in \Theta} \rho_n(x;\theta) \mathrm{d}\pi_n(\theta)\,, \tag{1}$$

avec

$$\rho_n(x;\theta) := \mathbf{E}_{n,\theta}((\xi(X_{n+1}) - M_n)_+ \mid X_{n+1} = x)$$

valeur de l'EI au point x sachant θ et $\mathcal{F}_n$, et $M_n = \xi(X_0) \vee \cdots \vee \xi(X_n)$. Si ξ est choisi gaussien, le critère ρ_n peut s'écrire sous la forme d'une expression analytique permettant de le calculer de façon efficace. Cependant, deux problèmes concernant la mise en œuvre se posent. a) Comment calculer une approximation de l'intégrale (1) ? b) De quelle façon procéder pour optimiser $\bar{\rho}_n$ à chaque étape ?

La plupart des mises en œuvre, au rang desquelles le célèbre algorithme EGO [3], traitent le premier problème en utilisant une approche empirique dite *plug-in* qui consiste à considérer comme approximation de π_n un Dirac dont la masse est concentrée au maximum de vraisemblance en θ. Une approche *plug-in* faisant appel au maximum *a posteriori* est également possible [6]. Les méthodes *complètement bayésiennes* sont plus difficiles à mettre en œuvre mais prometteuses (voir [4] et ses références). Concernant l'optimisation de $\bar{\rho}_n$ à chaque étape, plusieurs stratégies ont été proposées (voir, par exemple, [3, 5, 7, 10]).

Cet article traite les deux questions simultanément à l'aide d'une approche SMC (*sequential Monte Carlo*) [8, 9]. Les idées principales sont les suivantes. Dans un premier temps, comme dans [5], un ensemble pondéré $\mathfrak{T}_n = \{(\theta_{n,i}, w_{n,i}) \in \Theta \times \mathbb{R}, 1 \leq i \leq I\}$ échantillonné selon π_n est utilisé afin de calculer une approximation de $\bar{\rho}_n$; autrement dit, afin que $\sum_{i=1}^I w_{n,i}\ \rho_n(x;\theta_{n,i}) \to_I \bar{\rho}_n(x)$. De plus, à chaque étape n, on associe à chaque $\theta_{n,i}$ un (petit) ensemble de points candidats $\{x_{n,i,j}, 1 \leq j \leq J\}$ échantillonné selon une loi cible pertinente pour cette valeur particulière de θ (la probabilité d'amélioration, normalisée, peut être considérée comme un exemple de loi cible pertinente), de sorte que $\max_{i,j} \bar{\rho}_n\,(x_{n,i,j}) \approx \max_x \bar{\rho}_n(x)$.

2 Description de l'algorithme

À chaque étape $n \geq n_0$ de l'algorithme, l'objectif est de construire, dans $\Theta \times \mathbb{X}$, un ensemble de particules pondérées

$$\mathfrak{G}_n = \Big\{ \left(\gamma_{n,i,j}, w'_{n,i,j}\right) ,\ \gamma_{n,i,j} = (\theta_{n,i}, x_{n,i,j}) \in \Theta \times \mathbb{X}, w'_{n,i,j} \in \mathbb{R} ,\ 1 \leq i \leq I,\ 1 \leq j \leq J \Big\},$$

tel que $\sum_{i,j} w'_{n,i,j} \delta_{\gamma_{n,i,j}} \to_{I,J} \pi'_n$, avec

$$\mathrm{d}\pi'_n(\gamma) \ = \ \tilde{g}_n(x \mid \theta)\, \mathrm{d}\lambda(x)\, \mathrm{d}\pi_n(\theta)\,, \quad x \in \mathbb{X}\,,\ \theta \in \Theta\,,\ \gamma = (\theta, x),$$

où λ correspond à la mesure de Lebesgue, $\tilde{g}_n(x \mid \theta) = g_n(x \mid \theta)/c_n(\theta)$, $g_n(x \mid \theta)$ caractérise l'intérêt d'une évaluation en x (sachant θ et les observations précédentes), et $c_n(\theta)$ est un terme de normalisation valant $\int_{\mathbb{X}} g_n(x \mid \theta)\mathrm{d}x$. Un choix pertinent pour g_n est, par exemple, de considérer la probabilité que ξ soit supérieure à M_n au point x, à l'étape n. (Il est important de garder à l'esprit que l'on considère moins de valeurs de θ que de x dans $\mathfrak{G}_n$, et ce, afin de garder le coût algorithmique relativement faible.)

À l'initialisation, on génère un échantillon pondéré $\mathfrak{T}_{n_0} = \{(\theta_{n_0,i}, w_{n_0,i}),\ 1 \leq i \leq I\}$ à partir de la loi π_{n_0}, en utilisant, par exemple, de l'échantillonnage d'importance avec π_0 comme loi instrumentale, et on choisit une densité q_{n_0} sur $\mathbb{X}$ (densité uniforme, par exemple). Ensuite, pour $n \geq n_0$, l'algorithme procède en quatre étapes,

Étape 1 : démarginalisation — Utiliser $\mathfrak{T}_n$ et q_n pour construire un ensemble $\mathfrak{G}_n$, avec $x_{n,i,j} \overset{\text{iid}}{\sim} q_n$, $w'_{n,i,j} = w_{n,i} \frac{g_n(x_{n,i,j}|\theta_{n,i})}{q_n(x_{n,i,j}) c_{n,i}}$, et $c_{n,i} = \frac{1}{J} \sum_{j'=1}^{J} \frac{g_n(x_{n,i,j'}|\theta_{n,i})}{q_n(x_{n,i,j'})}$.

Étape 2 : évaluation — Évaluer ξ en $X_{n+1} = \mathrm{argmax}_{i,j} \sum_{i'=1}^{I} w_{n,i'}\, \rho_n(x_{n,i,j}; \theta_{n,i'})$. Cette étape est coûteuse en apparence, mais il est possible, en raffinant le calcul du critère, d'en obtenir une approximation satisfaisante pour un temps de calcul acceptable.

Étape 3 : pondération/échantillonnage/déplacement — Construire $\mathfrak{T}_{n+1}$ à partir de $\mathfrak{T}_n$: repondérer les $\theta_{n,i}$s avec $w_{n+1,i} \propto \frac{\pi_{n+1}(\theta_{n,i})}{\pi_n(\theta_{n,i})} w_{n,i}$, rééchantillonner (par exemple à l'aide d'un échantillonnage multinomial), et déplacer les $\theta_{n,i}$ pour obtenir les $\theta_{n+1,i}$ à l'aide d'un noyau de Metropolis-Hastings indépendant, comme dans [8].

Étape 4 : construire q_{n+1} — Construire une estimation q_{n+1} de la seconde loi marginale de π'_n à partir de $\mathfrak{X}_n = \{(x_{n,i,j}, w'_{n,i,j}), 1 \leq i \leq I, 1 \leq j \leq J\}$. Ce q_{n+1} correspond à une loi instrumentale qui permettra, lors de la prochaine étape de démarginalisation, de générer un exemple de nouveaux points candidats pertinents (relativement à la loi cible choisie pour les xs). Tout estimateur de densité (qu'il soit paramétrique ou non) peut être utilisé, à condition qu'un échantillonnage à partir de lui soit facile. Dans cet article, un estimateur à base d'arbres est utilisé.

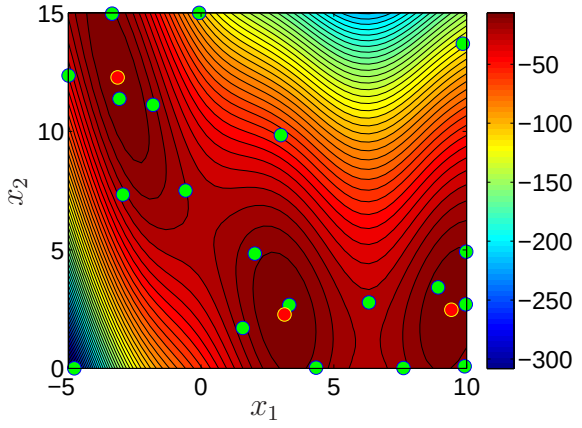
Nota bene : Lorsque cela est possible, certaines composantes de θ sont intégrées analytiquement dans (1) au lieu d'être échantillonnées ; voir [4].

3 Illustration

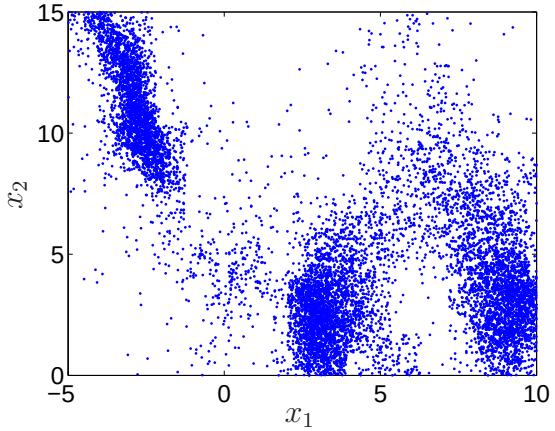
Les simulations sont réalisées à partir de la configuration suivante. On fait le choix d'un processus gaussien ξ avec une moyenne constante mais inconnue (à laquelle est

associée une distribution uniforme sur $\mathbb{R}$) et une covariance de Matérn anisotrope dont le paramètre de régularité ν vaut $5/2$. De plus, on utilise un *a priori* de Jeffreys sur le paramètre de variance de la covariance de Matérn. La moyenne et la variance peuvent être intégrées analytiquement. Les paramètres de portées, quant à eux, sont associés à des *a priori* lognormaux indépendants. On prend également $I = J = 100$, et on choisit la probabilité d'amélioration, normalisée, comme loi cible pour les xs.

On illustre ici le fonctionnement de l'algorithme sur une fonction test classique en dimension deux, la fonction de Branin (voir figure 1(a)). La figure 1(b) montre la répartition des points candidats sur le domaine après 20 évaluations. L'ensemble de la figure 1, permet ainsi d'observer qu'au voisinage des maxima, la densité de ces points est particulièrement importante, ce qui permet une maximisation efficace du critère EI (peu de points, mais répartis sur les zones particulièrement prometteuses). La figure 2(a), quant à elle, met en évidence la répartition des particules selon la loi *a posteriori* π_n. Ceci montre que l'approche utilisée ici permet d'apporter une réponse aux deux problèmes évoqués plus haut comme motivation à notre travail.



(a) Fonction Branin avec les maxima (en rouge) et les évaluations (en vert)

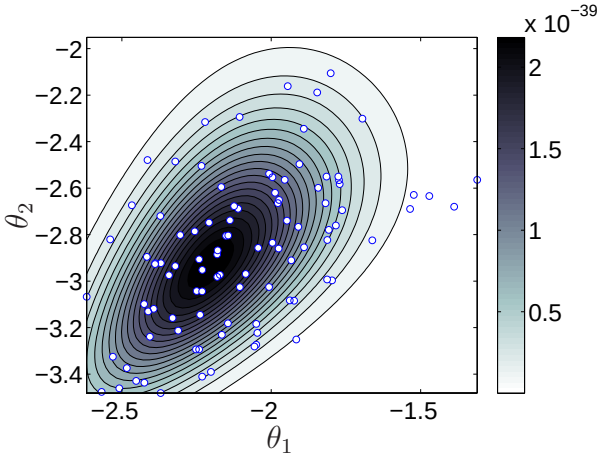


(b) Répartion des points candidats

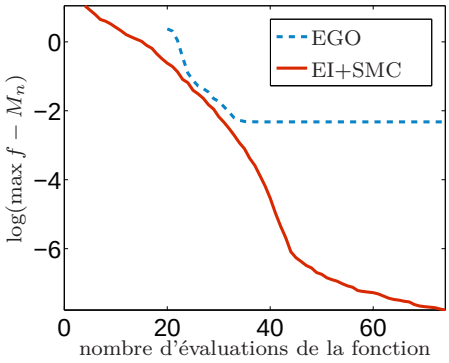
FIGURE 1 – Comparaison, après 20 évaluations, entre la fonction Branin et la répartition des points candidats

4 Comparaisons

Tests. Des résultats précédents montrant l'intérêt d'une approche complètement bayésienne par rapport à une approche empirique (dite *plug-in*), ont été établis dans [4]. Cependant, ces résultats doivent être tempérés dans la mesure où la méthode utilisée était relativement simpliste (elle faisait intervenir une approximation de $\bar{\rho}_n$ par quadrature et



(a) loi *a posteriori* $\pi_n(\theta)$ et répartition des particules en θ (après 20 évaluations)



(b) Comparaison de l'erreur moyenne au maximum (sur 100 expériences) pour la fonction Branin

FIGURE 2 – Illustration du fonctionnement de l'algorithme (répartition des particules en θ et comparaisons des performances en moyenne avec un algorithme de référence)

X_{n+1} était choisi sur une grille fixée). Nous présentons ici des résultats montrant que notre nouvel algorithme, basé sur des méthodes SMC, est capable de surmonter ces limitations.

L'idée est de comparer notre algorithme, avec les mêmes paramètres que lors de la partie précédente, à un algorithme de type EGO [3], où : 1) la valeur de θ est estimée à chaque itération par maximum de vraisemblance restreint ; 2) X_{n+1} est obtenu à l'aide d'une recherche exhaustive sur un LHS fixé de taille $I \times J$. Pour EGO, nous considérons également un processus gaussien ξ avec moyenne constante mais inconnue et une covariance de Matérn. Initialement, on se donne vingt points (dix fois la dimension) pour EGO, comme conseillé dans [3], afin d'avoir suffisamment de données pour qu'une estimation des paramètres par maximum de vraisemblance soit pertinente. Comme évoqué dans [6], un tel nombre de points initiaux n'est pas nécessaire lorsqu'un *a priori* est associé aux paramètres, c'est la raison pour laquelle seulement quatre points (deux fois la dimension) sont considérés initialement pour notre algorithme, ce qui permet de préserver ainsi le budget d'évaluation. Les points initiaux sont choisis selon un LHS, et renouvelés à chaque nouvelle expérience.

Résultats. Pour comparer la convergence de ces deux algorithmes, nous nous sommes intéressés à la moyenne de l'erreur au maximum (sur cent expériences), à nouveau pour la fonction Branin. La figure 2(b) montre que les performances en moyenne de notre algorithme complètement bayésien, à base de méthodes SMC, sont meilleures sur l'ensemble des itérations. Avoir choisi seulement quatre points d'évaluation pour l'initialisation est suffisant pour que l'algorithme optimise de façon efficace. En effet, on remarque que lorsqu'EGO commence à optimiser (à partir de 20 évaluations), l'erreur moyenne est déjà nettement moindre pour notre algorithme. Par ailleurs, on peut remarquer que la stratégie

de recherche non adaptative de l'algorithme EGO entraîne rapidement un décrochement important de sa part. L'ensemble des points candidats étant défini par un LHS fixé, EGO est tributaire de la précision optimale permise par ce LHS, et ne peut l'améliorer. Ceci n'est pas le cas de notre algorithme puisque l'ensemble des points candidats est regénéré entièrement après chaque évaluation, et se reparti sur les zones particulièrement prometteuses du domaine, permettant ainsi d'augmenter la précision au voisinage des maxima.

Sur cet exemple, les résultats indiquent clairement que l'utilisation d'*a priori* sur le paramètre θ, ainsi que la répartition, à chaque itération, de l'ensemble des points candidats selon une loi pertinente, permet de surmonter les difficultés de mise en œuvre inhérentes aux méthodes d'optimisation bayésienne. Notre algorithme s'impose donc comme une alternative solide face aux algorithmes classiques du domaine.

Références

[1] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. Dixon and G. Szego, editors, *Towards Global Optimization*, volume 2, pages 117–129. Elsevier, 1978.

[2] T. J. Santner, B. J. Williams, W. I. Notz. The design and analysis of computer experiments, Springer, 2003.

[3] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *J. Global Optim.*, 13(4) :455–492, 1998.

[4] R. Benassi, J. Bect, and E. Vazquez. Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion. In *LION5, online proceedings*, Roma, Italy, 2011.

[5] R. Gramacy and N. Polson. Particle learning of Gaussian process models for sequential design and optimization, *J. Comput. Graph. Stat.*, 20(1) :102–118, 2011.

[6] D. J. Lizotte, R. Greiner and D. Schuurmans. An experimental methodology for response surface optimization methods, *J. Global Optim.*, 38 pages, 2011.

[7] R. Bardenet and B. Kégl. Surrogating the surrogate : accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm. In *ICML 2010, proceedings*, Haifa, Israel, 2010.

[8] N. Chopin. A sequential particle filter method for static models, *Biometrika*, 89(3) :539–552, 2002.

[9] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers, *J. R. Stat. Soc. B*, 68(3) :411–436, 2006.

[10] D. Ginsbourger and O. Roustant. DiceOptim : Kriging-based optimization for computer experiments, R package version 1.2, 2011.